



SDP Workshop

Nashville TN – 13 Dec 2001

Architectures for Distributed Real-time Systems

Michael W. Masters – NSWCDD

Building Systems for the “Real” World

What is the Problem?

- Capability sustainment
- Affordable life cycle
- Low upgrade cycle time
- Technology refreshable despite obsolescence



Multi-dimensional Trade Space!

- | | |
|-------------|--------------------|
| • Real-time | • Distributed |
| • Complex | • Mission critical |
| • Long life | • Cost conscious |
| • Evolving | • High performance |

High Performance Distributed Computing

DARPA Goal:

Transition Computing
Technology to Military



*HiPer-D
Quorum*

HiPer-D Premise:

New Computer Program
& System Architecture
Required to Fully Exploit
COTS Technology

Navy Team

Architecture

Industry

Navy Goal:

Provide Increased
Capacity & Scalability



*Navy Real-time
Systems*

DARPA Technologies

- Advanced computers
- Operating systems
- Advanced networks
- Low latency protocols
- Quality-of-service middleware
- Resource management

Architecture Concepts

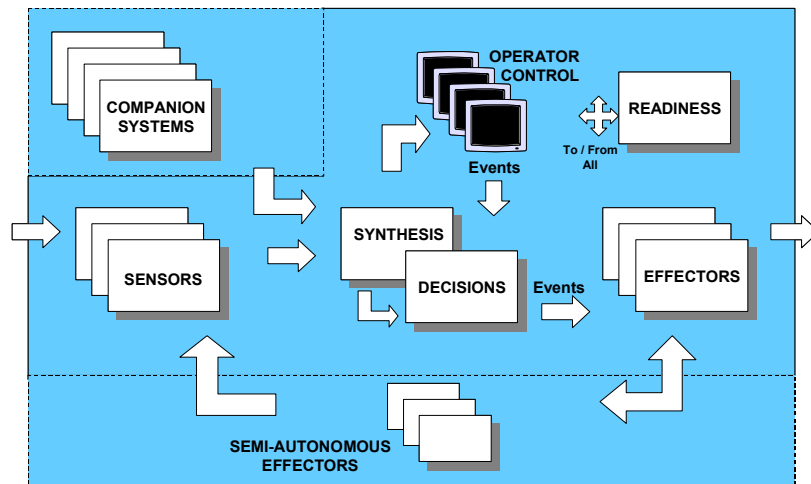
- Distributed processing
- Open systems
- Portability
- Scalability
- Fault tolerance
- Shared resource mgt.
- Self-instrumented

Navy Benefits

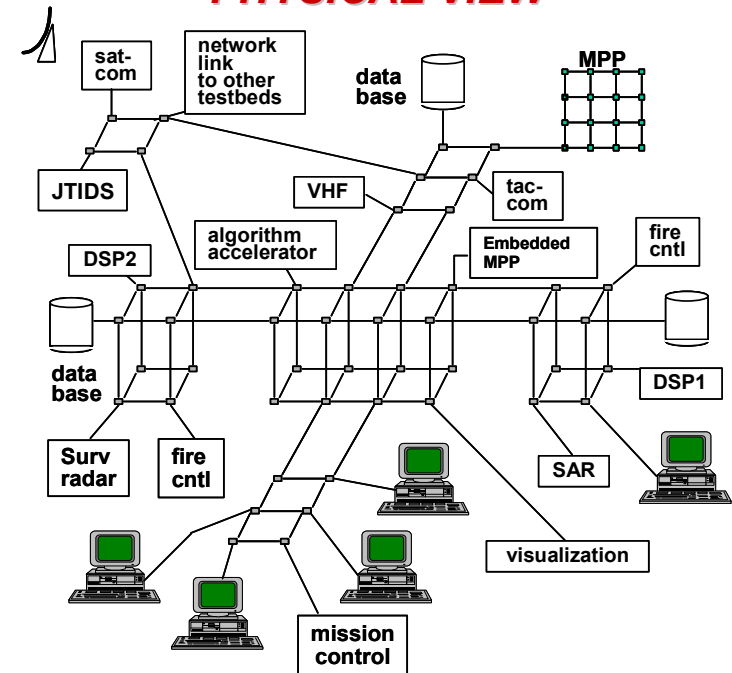
- Load-invariant tactical performance
- Information access
- Mission flexibility
- Continuous availability
- Rapid upgrades
- Low ownership cost

SYSTEM ARCHITECTURE VIEWS

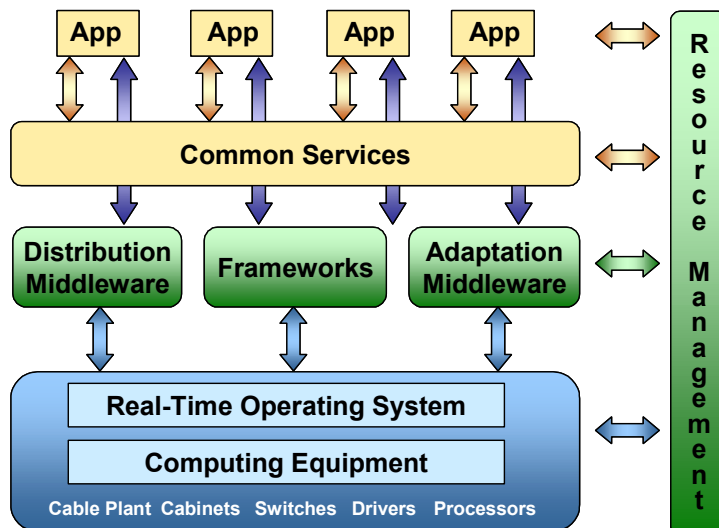
FUNCTIONAL VIEW



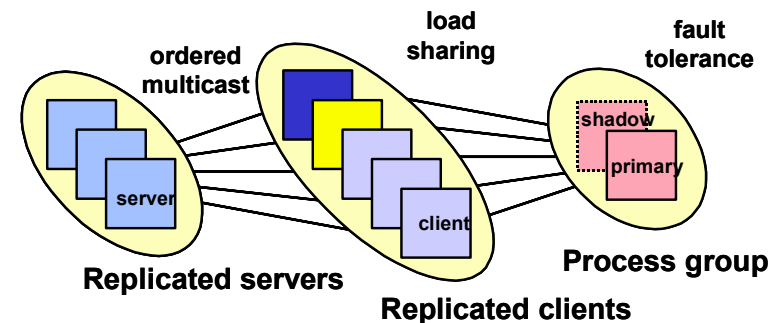
PHYSICAL VIEW



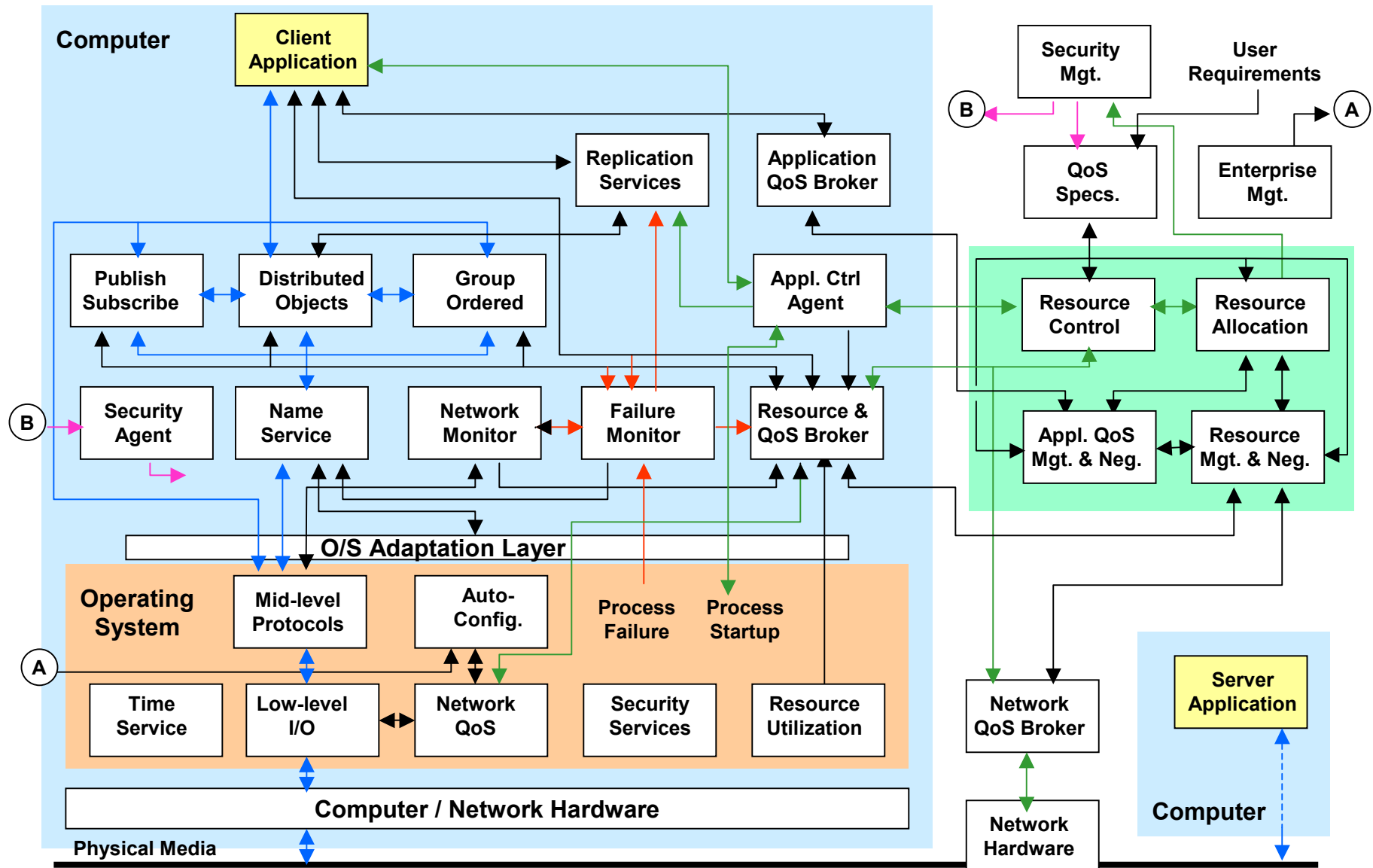
TECHNOLOGY VIEW



SOFTWARE VIEW



QoS REFERENCE ARCHITECTURE



GUIDANCE DOCUMENT

Computer Program Design

- Component partitioning
- Portability
- Location transparency
- Client-server
- Data distribution
- State data coherency
- Computational flow
- Fault tolerance
- Scalability
- Real-time performance
- Process, thread & memory mgt.
- Data flow management
- Track data distribution
- Legacy capture

Computing Technology Base

- Cabling and cabinets
- Information transfer
- Computing resources
- Peripherals
- Middleware
- Resource management
- Instrumentation
- Failure management
- Information assurance
- Time services
- Programming/language support facilities
- Requirements and design tools, methodologies and processes

CHALLENGES FOR THE FUTURE...

- **Fault Tolerance**
 - Faster fault detection and isolation ($\ll 1$ sec), e.g. via hardware support for fault detection and reconfiguration
 - Integrated failure management across technology base
- **Middleware**
 - Faster, scalable performance during join, leave & recovery events
 - Integrated products with full range of middleware functionality
 - Middleware for higher performance domains
- **Resource Management**
 - Optimal, stable system-wide dynamic allocation algorithms
 - Run-time schedulability and stability analysis for mixed real-time systems (hard, soft, event)
 - Incorporation of network QoS and routing management
- **Security**
 - Intrusion detection, authentication, mgt. of security domains, etc.
 - Integration with other technologies, e.g. Resource Management
- **System**
 - Support for system end-to-end performance requirements
 - Certification methods for dynamically allocated systems